

Application No.: 10/786,748
Reply to FOA date: February 1, 2007
Reply dated: April 2, 2007

Remarks

This REPLY and these Remarks are in response to the Final Office Action mailed February 1, 2007. Applicant believes no additional fee is due with this communication.

I. Summary of Examiner's Rejections

Prior to the Office Action mailed February 1, 2007, Claims 1, 4, 6-8, 10, 11, 14, 16-18, 20, 21, 24, 26-28 and 30-33 were pending in the Application. In the Office Action, the Specification and Claims 1, 11 and 21 were objected to for various informalities. Claims 1, 4, 6-8, 10, 11, 14, 16, 18, 20, 21, 24, 26, 28 and 30 were rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al. (U.S. Publication No. 2004/0088681, hereinafter Berg) in view of Rich et al. (U.S. Publication No. 2002/0178439, hereinafter Rich). Claims 7, 17, 27 and 31-33 were rejected under 35 U.S.C. 103(a) as being unpatentable over Berg in view of Rich, and further in view of McIntyre (U.S. Patent No. 6,178,546).

II. Summary of Applicant's Amendments

The present Reply amends the Specification; and Claims 1, 11, 21 and 31-33, leaving for the Examiner's present consideration Claims 1, 4, 6-8, 10, 11, 14, 16-18, 20, 21, 24, 26-28 and 30-33.

III. Objections to the Specification

In the Office Action mailed February 1, 2007, the Specification was objected to for various informalities. Accordingly, the Specification has been amended as shown above. Applicant respectfully submits that the proposed amendments correct informalities in the Specification and that no new matter is being added.

IV. Objections to the Claims

In the Office Action mailed February 1, 2007, Claims 1, 11 and 21 were objected to for various informalities. Accordingly, the claims have been amended as shown above to correct the informalities. Reconsideration thereof is respectfully requested.

V. Claim Rejections under 35 U.S.C. §103(a)

In the Office Action mailed February 1, 2007, Claims 1, 4, 6-8, 10, 11, 14, 16, 18, 20, 21, 24, 26, 28 and 30 were rejected under 35 U.S.C. 103(a) as being unpatentable over Berg (U.S. Publication No. 2004/0088681) in view of Rich (U.S. Publication No. 2002/0178439). Claims 7, 17, 27 and 31-33 were rejected under 35 U.S.C. 103(a) as being unpatentable over Berg in view of Rich, and further in view of McIntyre (U.S. Patent No. 6,178,546).

Claim 1

Applicant respectfully traverses the rejection of Claim 1. As presently defined, Claim 1 defines an embodiment that comprises a split directory structure which includes both a source folder that stores editable source files as part of the software application, and a corresponding output folder that stores compiled files as part of the software application, and wherein the split directory is accessed as a virtual file that provides an abstraction over the two folders therein. Claim 1 further defines that during deployment the server deploys the application by making requests to the virtual file which checks both the source folder and the corresponding output folder for software application files, before deploying the software application files to the server.

The advantages of the embodiment defined by Claim 1 include that, during deployment of the software application, the source and output folder are interpreted as a single folder or directory. This approach requires no copying, in that the server can read source files (for example JSP's, XML descriptors, html images, etc.) directly from the split directory structure, without having to first copy them to a build directory. The server receiving the build can see both the /build folder, and the /source folder. This allows, for example, Web files to be changed and redeployed in place within the source folder, without having to rebuild the entire software application.

Berg discloses a method and system for dynamically mapping archive files in an enterprise application. As disclosed therein, the highest-level project, (the project that "contains" the nested archives, or contains references to other projects that conceptually represent nested archives) is referred to generically as a "container project". Two examples of container projects in a J2EE environment are EAR projects, that represent enterprise applications or EAR files, and Web

projects, that represent web applications or WAR files. The application server, however, expects the files to be in a hierarchical directory structure as noted above. (Paragraph [0010]). The developer begins by either developing a new container project or by using an existing container project. As is well known, the container project will typically identify multiple referenced projects and direct their interaction between each other. Each referenced project is mapped to a URI including a file name and possibly one or more directory/subdirectories. (Paragraph [0025]).

However, Applicant respectfully submits that, in Berg, each of the projects within the container project appear to include previously compiled code, including WAR and JAR archives, so that the container project as a whole contains a hierarchy of previously compiled projects and archives. Claim 1 instead defines that the split directory structure includes both a source folder that stores *editable source files* as part of the software application, and a *corresponding output folder that stores compiled files* as part of the software application.

Rich discloses a method and system for providing a programming interface for loading and saving archives in enterprise applications. As disclosed therein, the two environments, assembly and "runtime" (the running application server), are therefore dealing with two different physical file structures: assembly, with JARs; and runtime, with a mixture of expanded JARs in a directory tree structure and JAR files themselves. There is much commonality between the two environments, however; specifically the need to load, edit, and save deployment information. What is needed is a library of programming APIs (application programming interface) that can be shared between the two systems so that a user (e.g., a programmer) can load, edit/manipulate, and save files using one set of commands without having to know which structure, archive or directory tree is in use (Paragraph [0014]). At step 102, the user (running program) requests the loading of an archive having a given file path/name. At step 104, a determination is made if the requested file is an archive file type (e.g., .jar, .zip, .war, etc.) or a directory tree file. At step 106, a loading strategy for loading and displaying the file requested by the user ("open archive" command) is created based on the status of the file requested as having an archive structure or a directory tree. At step 108, a "virtual" archive to return to the calling method is created based upon the loading strategy. (Paragraphs [0025-0028]).

In the Office Action it was submitted that the virtual archive of Rich is analogous to the virtual file defined in Claim 1. However, Applicant respectfully submits that, in Rich, it appears that

Application No.: 10/786,748
Reply to FOA date: February 1, 2007
Reply dated: April 2, 2007

the user (i.e. the running program) requests the loading of an archive, and the system then uses the virtual archive to load, edit/manipulate, and save files using one set of commands, without having to know which structure, archive or directory tree is in use. As such, Rich appears to be directed to allowing users to access archives at runtime, subsequent to deployment. Claim 1 instead defines that *during deployment* the server recognizes the split directory structure and *deploys the application by making requests to the virtual file* which checks both the source folder and the corresponding output folder for software application files, before deploying the software application files to the server.

In view of the above comments, Applicant respectfully submits that Claim 1, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claims 4, 6-8, 10, 11, 14, 16-18, 20, 21, 24, 26-28 and 30-33

Claims 4, 6-8, 10, 11, 14, 16-18, 20, 21, 24, 26-28 and 30-33 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Reconsideration thereof is respectfully requested.

VI. Conclusion

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Application No.: 10/786,748
Reply to FOA date: February 1, 2007
Reply dated: April 2, 2007

Applicant believes no additional fee is due with this communication. However, the Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: April 2, 2007 By: /Karl F. Kenna/
Karl F. Kenna
Reg. No. 45,445

Customer No.: 23910
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800